Amendment To The Claims

1. (Original) A system for offloading TCP processing, the system comprising:

a host;

a network interface card (NIC) coupled to said host, said NIC comprising,

a TCP enabled Ethernet controller (TEEC), said TEEC comprising,

at least one internal elastic buffer, wherein said TEEC processes an incoming TCP packet once and temporarily buffers at least a portion of said incoming TCP packet in said internal elastic buffer, said processing occurring without reassembly.

2. (Original) The system according to claim 1, wherein said at least one internal elastic buffer comprises at least one of a receive internal elastic buffer and a transmit internal elastic buffer.

3. (Original) The system according to claim 2, wherein said at least a portion of said incoming TCP packet is temporarily buffered in said receive internal elastic buffer.

4. (Original) The system according to claim 2, wherein at least a portion of a TCP packet to be transmitted is temporarily buffered in said transmit internal elastic buffer.

5. (Original) The system according to claim 1, wherein said TEEC places at least a portion of said incoming TCP packet data into at least a portion of a host memory.

6. (Original) The system according to claim 1, wherein said NIC utilizes only said at least one internal elastic buffer to temporarily buffer said at least a portion of said incoming TCP packet.

7. (Original) The system according to claim 1, wherein out-of-order TCP packets are not at least one of stored, re-ordered and re-assembled in a TEEC buffer.

8. (Original) The system according to claim 1, wherein said NIC does not require a dedicated memory for re-ordering out-of-sequence TCP packets.

9. (Original) The system according to claim 1, wherein said NIC does not require a dedicated memory for assembling and re-ordering IP packets fragmented at the IP layer.

10. (Original) The system according to claim 1, wherein said TEEC places at least data from said incoming TCP packet into a highest hierarchy of buffer available in a host memory by performing a single copy operation.

11. (Original) The system according to claim 1, wherein said TEEC DMA transfers at least a portion of said processed incoming TCP packet to at least a portion of a host memory.

12. (Currently amended) The system according to claim 1, wherein said NIC does not require a TCP offload engine (TOE) ~~TOE~~ dedicated memory for at least one of packet retransmission and packet reassembly.

13. (Original) The system according to claim 1, wherein said TEEC places at least a portion of said processed incoming TCP packets into host buffers in a host memory for reassembly.

14. (Original) The system according to claim 1, wherein said TEEC comprises a single chip, having integrated therein, said at least one internal elastic buffer.

15. (Original) The system according to claim 1, wherein said TEEC comprises a single chip, having integrated therein, said at least one internal elastic buffer, and no internal buffers and interfaces to external buffers, that are utilized for at least one of packet retransmission, packet reassembly and packet re-ordering.

16. (Currently amended) A method for offloading TCP processing, the method comprising:

receiving an incoming TCP packet at a TCP enabled Ethernet controller (TEEC) ~~TEEC~~;

processing at least a portion of said incoming packet once by said TEEC without reassembly; and

temporarily buffering said at least a portion of said incoming TCP packet in at least one internal elastic buffer of said TEEC.

17. (Original) The method according to claim 16, wherein said at least one internal elastic buffer comprises at least one of a receive internal elastic buffer and a transmit internal elastic buffer.

18. (Original) The method according to claim 17, further comprising temporarily buffering said at least a portion of said incoming TCP packet in said receive internal elastic buffer.

19. (Original) The method according to claim 16, further comprising placing at least a portion of said processed at least a portion of said incoming packet in at least a portion of a host memory.

20. (Original) The method according to claim 16, wherein said placing further comprises placing at least a portion of said processed incoming TCP

packet in a highest hierarchy of buffer available in a host memory by performing a single copy operation.

21. (Original) The method according to claim 16, further comprising DMA transferring at least a portion of said processed incoming TCP packet in at least a portion of a host memory.

22. (Original) The method according to claim 16, wherein packets temporarily buffered in said in at least one internal elastic buffer are not buffered for at least one of reassembly and retransmission.

23. (Original) The method according to claim 16, further comprising placing at least a portion of said processed incoming TCP packet in host buffers in a host memory for processing.

24. (Original) The method according to claim 16, wherein said TEEC comprises a single chip, having integrated therein, said at least one internal elastic buffer.

25. (Currently amended) A machine-readable storage, having stored thereon, a computer program having at least one code section for providing TCP offload, the at least one code section being executable by a machine for causing the machine to perform steps comprising:

receiving an incoming TCP packet at a TCP enabled Ethernet controller (TEEC) ~~TEEC~~;

processing at least a portion of said incoming packet once by said TEEC without reassembly; and

temporarily buffering said at least a portion of said incoming TCP packet in at least one internal elastic buffer of said TEEC.

26. (Original) The machine-readable storage according to claim 25, wherein said at least one internal elastic buffer comprises at least one of a receive internal elastic buffer and a transmit internal elastic buffer.

27. (Original) The machine-readable storage according to claim 26, code for temporarily buffering said at least a portion of said incoming TCP packet in said receive internal elastic buffer.

28. (Original) The machine-readable storage according to claim 25, further comprising code for placing at least a portion of said processed at least a portion of said incoming packet in at least a portion of a host memory.

29. (Original) The machine-readable storage according to claim 25, further comprising code for placing at least a portion of said processed incoming TCP packet in a highest hierarchy of buffer available in a host memory by performing a single copy operation.

30. (Original) The machine-readable storage according to claim 25, further comprising code for DMA transferring at least a portion of said processed incoming TCP packet in at least a portion of a host memory.

31. (Original) The machine-readable storage according to claim 25, wherein packets temporarily buffered in said in at least one internal elastic buffer are not buffered for at least one of reassembly and retransmission.

32. (Original) The machine-readable storage according to claim 25, further comprising code for placing at least a portion of said processed incoming TCP packet in host buffers in a host memory for processing.

33. (Original) The machine-readable storage according to claim 25, wherein said TEEC comprises a single chip, having integrated therein, said at least one internal elastic buffer.